

S32-37

88

118462

# A Flexible Telerobotic System for Space Operations

N.O. Sliwa and R.W. Will  
NASA Langley Research Center  
Hampton, VA 23665-5225

ND21047

## 1. Abstract

Robotic systems will play an increasingly important role in space operations. This paper describes the objective and design of a proposed goal-oriented knowledge-based telerobotic system for space operations. This design effort encompasses the elements of the system executive and user interface, and the distribution and general structure of the knowledge base, the displays, and the task sequencing.

The objective of the design effort is to provide an expandable structure for a telerobotic system that provides cooperative interaction between the human operator and computer control. The initial phase of the implementation provides a rule-based, goal-oriented script generator to interface to the existing control modes of the telerobotic research system in the Intelligent Systems Research Lab at NASA Langley Research Center.

## 2. Introduction

The hostile environment of space encourages the use of advanced mechanical and computational means to assist man's development of space as a productive environment. The use of teleoperated manipulators, such as the Shuttle's Remote Manipulator System, has proven the practicality of this concept, both in space and in other hostile environments such as undersea and in nuclear reactor complexes.

The proposed Space Station represents a completely new approach to space operations. On-orbit assembly and servicing operations will expand in number, duration, and complexity to exceed the capability of manned extravehicular activity. The Space Station must have an intelligent remote manipulation system to assist astronauts in both assembly and maintenance tasks.

This paper describes the objective and design elements of a proposed goal-oriented telerobotic system for space operations. It also describes the initial implementation phase of this system in the Intelligent Systems Research Lab (ISRL) at NASA Langley Research Center.

## 3. Objective

A remote manipulation system for Space Station operations will be called upon to perform a wide variety of sophisticated tasks on a Space Station whose configuration will evolve. A manipulator system must interact with advanced systems and with modules whose form and function are not yet defined. The system should operate with minimum human intervention, at least for routine or prolonged operations. The Space Station remote manipulation system must therefore be generalized, flexible, and easily reprogrammable to accommodate the wide range of tasks. Very high-level task-oriented command capability is required to provide such features. The only way to support task-level commands is through internal knowledge which supplies the details not provided by the operator. The Space Station therefore requires a task or goal-oriented, knowledge-based remote manipulation system.

Such a system is considerably beyond the state-of-the-art. Since it is impossible to plan for all contingencies, any attempt to provide near-complete autonomy will result in an extremely cumbersome system which will necessarily fail in unexpected situations. A more acceptable concept of a system which cooperates closely with a human operator at increasingly sophisticated levels of command. Due to technological advances in processors, manipulator subsystems, and sensor subsystems, telerobotic systems are now possible in which control is shared between a human operator and knowledge-based software systems. Therefore, the objective of the design effort is to provide an evolutionary structure for a telerobotic system, i.e., one that can progress from strictly teleoperated through phases of serving as an assistant, a colleague, and an expert, to eventually serve as a relatively autonomous unit, requiring only minimal supervision. The system should be configured to encompass a full range of operational capabilities from basic teleoperation to nearly-autonomous performance involving task planning and status feedback.

#### 4. Design Issues

##### 4.1 Evolutionary Capability

A requirement for a goal-oriented telerobotic system for Space Station is that it be designed to accommodate new technology, both in hardware and software engineering, as it is developed. Such an evolution would allow the following roles to be assumed by a telerobotic system:

1. Advisor - Upon operator request, provides suggestions for operation sequences and guidance in performing tasks.
2. Monitor - System monitors progress and performance of teleoperator and provides warnings and possibly overrides dangerous actions.
3. Assistant - Operator requests system to perform tedious operations by initializing state and commanding desired actions (from one state to another). Operator must also be able to reassume control at any point or at certain states.
4. Planner - System plans and executes operations in response to task requests and with fixed scenarios. System appeals to operator when anomalies arise.
5. Expert planner - Same as planner except that plans are synthesized from generic subtasks and alternates are logically derived.
6. Robust planner - Same as expert planner except that alternates are developed dynamically in response to feedback and performance; and operator assumes control as last resort.
7. Learning planner - The system is able to autonomously improve its performance by generalizing circumstances of previous tasks.

The designs of the telerobotic system must keep in mind the ultimate objectives of the system performance in order to build an expandable structure.

##### 4.2 Shared Control

A key issue in the design of this system is the concept of "mixed initiative", i.e., control is flexibly shared between the human operator and the software system. This allows the human to monitor and assist the software system in task performance, and vice versa. This approach would increase system reliability and flexibility as well as increasing the credibility of its actions to the human operator. This type of control manifests itself in several ways to an operator, for example:

1. The software system, in performing an automated task sequence, may request the operator to perform a specific sequence of the task that itself is insufficiently sophisticated to perform.
2. The human operator can assist the software system in overcoming positional errors due to sensor and knowledge base uncertainty.
3. The human operator can manually override any gross errors by the software system due to catastrophic subsystem failure.
4. The software system can unload from the human operator well-defined but tedious subtasks, such as nut/bolt mating.
5. The software system can monitor the human operator's inputs and warn of any imminent operator errors.

These examples are intended to illustrate several different levels of sophistication of the system. The means of maintaining both the operator awareness of the current task situation, and the integrity of the software system knowledge base, is an important research issue.

##### 4.3 Symmetrical Structuring

A cooperative, mixed initiative approach will increase the complexity of the system beyond that of a strictly autonomous or a strictly teleoperated system. This is due to the need to share large amounts of information between the system and the human operator. This additional complexity impacts the system in several different ways, particularly (1) the initial development of the system, (2) the maintenance and expansion of the system, and (3) the use of the system by the operator.

To manage this additional complexity, the design of the system must be highly modular, both horizontally (i.e., hierarchical modularity) and vertically (i.e., functional modularity). Moreover, it is critical that each module be as symmetrical as possible, both in control and interface structure. For example, the displays for both the task planning and the knowledge bases use the same structure and are accessed and manipulated similarly. Likewise, the system executive uses symmetrical structure in the implementation of user commands within various modes.

This approach requires less software to be developed and maintained, and permits an easier understanding of the total system by the programmers. It also makes the user interface easier for the operator to learn and remember.

## 5. Basic System Structure

Figure 1 is a block diagram of a goal-oriented telerobotic system incorporating several types of internal knowledge, sensor feedback, and multiple levels of command access. It is divided logically into hierarchical levels which support system evolution and the "mixed initiative" concept. The system of figure 1 operates on three conceptual levels: teleoperator, programmed robot, and task executive, which includes planning and scheduling operations. At the teleoperator level, detailed spatial motion commands are supplied by the operator. At the robot programming level, a robot-level programming language mechanizes more generalized actions with respect to objects in the environment; i.e., ACQUIRE BOLT 26. The upper level of the goal-oriented telerobotic system, the task executive, is a task planner which uses facts, rules, and assembly details to synthesize robot programming language command sequences to accomplish complex tasks. Most human operator commands will eventually be at the task level, with the robot programming level used mainly for monitoring and debugging.

Since the goal-oriented telerobotic system is intended to be a generalized structure to support operations for Space Station assembly and servicing, it is hierarchical and modular. The lower levels may be developed first and then used as a foundation for the development of increasingly sophisticated upper levels. The mixed initiative concept permits development to proceed while maintaining system credibility with the human operators.

## 6. Component Designs

This section describes the individual subsystems and their internal functions in more detail. The emphasis is on the information and techniques used by each subsystem in the performance of the basic task execution, regardless of whether the operator or the system is providing active control.

### 6.1 System Executive

The system executive and man-machine interface are necessary components of the total system. These elements are shown in figure 2, which also shows the type of information communicated between the various subsystem elements.

The system executive defines the character of the entire system. Its primary function is to implement the mode control for the "mixed initiative" scheme, allowing either teleoperated control or computer control. The executive provides the primary user interface and is thus responsible for command processing and display selection. There are two types of commands available to the user within the computer control mode, execution commands and interrogation commands. With execution commands, the user controls task performance by the telerobotic system. The displays associated with execution commands are concerned with monitoring system performance at particular operational levels and in different levels of detail. With interrogation commands, the user can request displays of the structure and content of internal knowledge, displays of subsystem states, and displays of performance analysis.

System configuration and resources also vary depending on the control mode. Thus, another function of the system executive is to manage resources and resolve conflicts while coordinating prioritized subsystem operations and communications. Figure 2 indicates the communication interfaces and the types of information passed between the subsystem elements. Execution commands go to the task planner, the robot programming language, and the telerobotic hardware subsystems which return performance monitoring data. The remainder of the communication paths are for the exchange of information necessary for the functioning of subsystems in the performance of a task. Most of these are transfers of information from knowledge bases to the command processing subsystems.

### 6.2 Knowledge Bases

The intelligence of the software system behavior is directly proportional to the richness of the knowledge base. A primary research consideration is the representational structure and distribution of such a potentially complex knowledge base. This design effort explores the use of both a horizontal and a vertical distribution of system knowledge. Horizontally, each knowledge base module is distributed in a hierarchically abstracted tree structure. Vertically, the knowledge is divided into three units: (1) a System Knowledge Base, dealing with actuator and sensor states of the robotic devices; (2) a World Model Knowledge Base, dealing with objects and various other environmental attributes; and (3) a Task Knowledge Base, dealing with the interaction of the system with the environment in performing a specific task.

System Knowledge Base. The system knowledge base maintains the sensor/actuator data from the robotic system hardware. This data structure is used as a mailbox for communications between the actual robotic devices and the software system. Sensor data from the sensor hardware is continually fed into this mailbox, to be accessed by the higher-level control processes. Likewise, commanded actuator states from the higher-level control processes are placed in the mailbox to be delivered to the actuator hardware. This information is globally available, but is used primarily by the robot programming language, which needs position and status, and by the operator displays, to communicate system performance.

**World Model Knowledge Base.-** The world model knowledge base contains the system's perception of its operating environment and how to manipulate it. The knowledge must include: physical details such as dimensions, inertias, and color; locations and orientations; features such as grasp points and holes; descriptions of interconnections; and a physical and logical representation of the structure of complex objects. Figure 3 shows the structure of the world model knowledge base and lists the definitions of the terms used. The actual information, such as identifier, location and dimensions, is stored within data structures describing the members of the various lists. This information is supplied by the sensor subsystems and CAD/CAM data on the manufacture and assembly of the objects in the environment. The world model information is supplied to the task planner, the robot programming language, and on request, to the operator interface displays. The world model knowledge base is also the repository for various world states expected to exist at specified points of task performance and used to verify proper execution.

**Task Knowledge Base.-** In the performance of a task with a telerobotic system, there will be knowledge that deals specifically with the interaction of the system and the environment for performing the specific task. This class of knowledge has been modularized into the task knowledge base. Figure 4 shows a conceptualization of the three knowledge base modules. The task module would tend to deal with elements of the other knowledge bases at a higher level of abstraction, and would itself tend to generate more abstracted elements of knowledge. Part of this task-specific knowledge must be resident to plan and perform the task, while part will be created during the execution of the task. Most of this "created" knowledge will be transient and discarded by the system. However, in advanced systems, part of the created knowledge would be absorbed as a more permanent portion of the task knowledge base, giving the planner the ability to learn.

### 6.3 Programming Language

**TRICCS.-** A useful robot programming language should provide the ability to issue commands with respect to objects in the environment. The language executor would look these up in the knowledge base to translate the instructions to spatial coordinates. The TeleRobotic Integrated Command and Control (TRICCS) language [1] is an object-directed, higher-level command language for telerobotic systems which interacts with the knowledge base and serves as a target language for the task planner. Tasks would be decomposed into TRICCS command sequences, which would be further broken down into point-to-point spatial commands to the system actuators. TRICCS programs could also be written manually, but this would probably not be cost effective except for checkout and debugging purposes. TRICCS would include a subroutine capability to make the language extensible by encapsulating frequently used functions. The TRICCS interpreter would therefore contain extensive debugging facilities such as step execution and a conditional execution mode where a simulation rather than the robot hardware executes the commands.

**ISRL MENU.-** As the ISRL telerobotic research system has developed, a command language menu for this system has evolved. This menu has grown to have numerous commands, each requiring specific submenu sequences of varying depths. These menu options include the ability to select control via any combination of multiple sensors and actuators, and to define coordinate transforms pertinent to the environment.

The robot-level programming language, either TRICCS or the ISRL MENU language, will provide an invaluable monitoring and checkout aid. The decomposed task commands will be available to the human operator in a readable form. These commands are at a low level of abstraction, i.e., a high level of detail. This code may be displayed to the operator prior to execution to check program logical structure. It may also be displayed during execution to provide a trace of the command sequencing.

### 6.4 Task Planner

Task planning must be performed for all stages of evolution (section 4.1). For the advisor, the task planner output is simply displayed to the human operator in a step-by-step form, with the ability to respond to clarification queries. The monitor mode requires more analysis feedback from various sources to determine if system and world states and transitions are proceeding according to the task plan. In addition, it must decide if the operator's actions are potentially detrimental, warn the operator of the circumstances, possibly override the operator and explain why.

In the assistant mode, the operator may elect to transfer active control either to or from the system at specified points or states. In accepting control from the operator, the system must assess the world model and system knowledge bases, comparing them with the specified states to insure that it knows how to proceed. If there is no match, the differences must be portrayed to the operator and an exchange of information must occur in order to resolve the differences. Enough state and task information must be displayed to make sure that the operator understands the situation.

The various advanced planner modes (expert, robust, learning) do not involve significant involvement by the human operator. The system appeals to the operator for information or active help only when anomalies arise. This should occur less for higher modes where more alternatives are available and are more tailored to the situation. The operator will, however, always be able to ask "what are you doing?" and "why are you doing that?" types of questions.

### 6.5 Displays and Command Interfaces

The form of the displays is designed to be symmetric throughout the system and to offer varying degrees of detail depending on whether a task is currently executing or not. Displays of both the knowledge base modules and the task sequencing modules are displayed in a tree/lattice form, and command/feedback communications are displayed dynamically in sequenced text strings.

The primary objective of the operator interface is to transfer large amounts of sometimes complex information in a fast, easily understandable form. It should make maximum use of graphics since humans communicate more efficiently in this form; they are capable of absorbing larger amounts of information more quickly through graphic form than from simple text strings.

The structure of the internal data is represented by a hierarchical tree structure (fig. 5) for displaying the task planning information. Tasks and subtasks are successively divided into smaller and more detailed operations. The operator can traverse the tree and examine the sequence of operations at any level of detail. A similar tree structure may be developed for the world model information by substituting objects and parts for tasks and subtasks (see fig. 3) to examine the relationship and details of the environment at several levels.

## 7. Implementation

Given the complexity of the total planner design, and the limited resources for implementing the design, it was decided to use a phased development process. The initial phase of the development will consist of building a rule-based expert system to generate a script of ISRL MENU components. This script will be available to the operator as an advisor on how to perform a specific task. It will also be able to serve as input directly to the telerobotic system for execution.

Figure 6 shows the integration of this proposed planner with the current telerobotic system implementation. Figure 6a shows the current implementation: the operator has access to both the telecontrols and the menu options to control the system in performing a task. The proposed planner would add a third option to the operator (fig. 6b) and would plan task performance based on the capabilities of the other two control modes.

This initial implementation will be used to explore several issues. A primary issue is the expandability of the system. The ISRL MENU is continually changing. Options are expanded or condensed to reflect both new capability development and the elimination of obsolete commands. Methods will be explored for facilitating these changes within the planner.

Another issue is the transfer of control between operator and planner. This proposed implementation will have the ability to exchange control with the operator at specific points. This exchange interface will be exercised in various tasks to determine additional desirable capabilities.

A third issue to be explored by this implementation is the amount of feedback required within the system for task planning. Direct access to the system knowledge base, as well as to the task and world knowledge bases, may be required by the planner. This is expected to depend on the degree of operator interaction required by the planner.

## 8. Summary

A promising method of developing a robotic system for the Space Station involves a phased "mixed initiative" technique which provides a gradual transition from teleoperated to nearly-autonomous task performance. This approach provides flexibility, redundancy, on-board checkout, and operator credibility. This approach also lends itself to the use of modular and hierarchical structure in the design of all elements of the system, including the knowledge bases, the operator displays, and the task planning. This symmetry of design structure can decrease the complexity of the total system, making it easier to develop, maintain, and operate.

The goal-oriented telerobotic system described is in the early stages of development. The component and manipulator levels (fig. 1) have been initially implemented both in simulation [2], and in hardware [3]. A robot programming language has been defined and described [1]. A world model knowledge base has been designed and a report is in progress. An initial system knowledge base has been implemented and is undergoing refinement.

An initial task planner implementation is being developed. This implementation consists of a rule-based expert system which generates scripts of menu elements to interface to the existing ISRL telerobotic system implementation. This initial planner will be used to explore planner expandability, control exchange between operator and planner, and feedback/operator interaction requirements.

## 9. References

- [1] R.W. Will, "TRICCS: A Proposed Teleoperator/Robot Integrated Command and Control System Applications," NASA Langley Research Center, TM-87577, July 1985.
- [2] J.E. Pennington, "A Rate Controlled Teleoperator Task with Simulated Time Delays," NASA Langley Research Center, TM-85653, September 1983.
- [3] F.W. Harrison, and J.E. Pennington, "System Architecture for Telerobotic Servicing and Assembly Tasks," presented at the SPIE 1986 Cambridge Symposium on Optical and Optoelectronic Engineering, Cambridge, Massachusetts, October 1986.

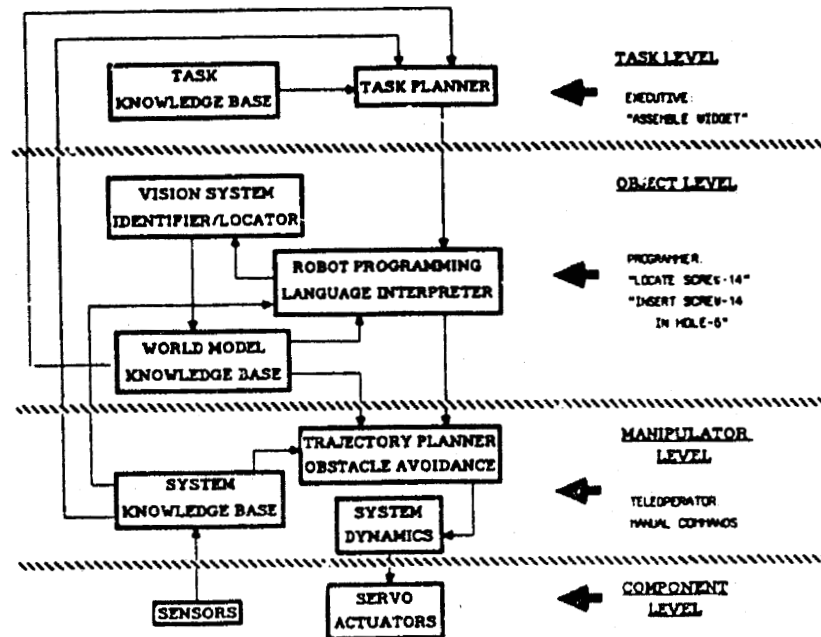


Figure 1 - Goal-oriented telerobotic system

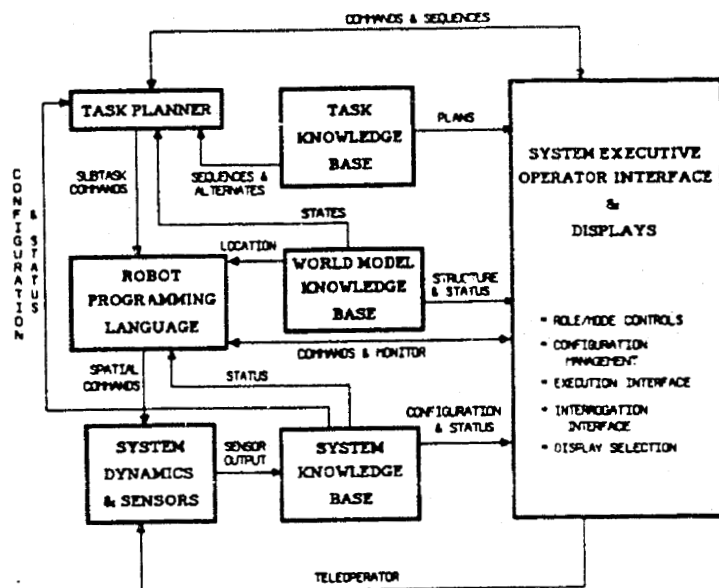
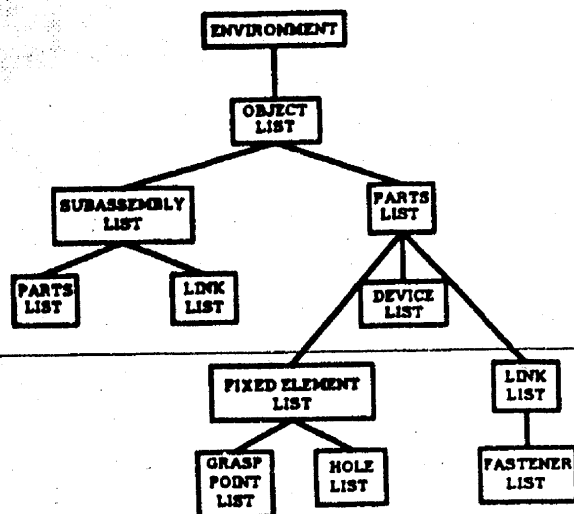


Figure 2 - Executive, operator, and display interface



**OBJECT** - SEPARATE, RELOCATABLE ENTITY WITH NO PHYSICAL CONNECTIONS

**PART** - MOVABLE OR REMOVABLE, INDIVISIBLE ENTITY ATTACHED TO OBJECTS OR SUBASSEMBLIES

**FIXED ELEMENT** - IMMOVABLE, UNREMOVABLE STRUCTURE OF A PART

**LINK** - MECHANICAL CONNECTION BETWEEN TWO PARTS

**SUBASSEMBLY** - LOGICAL GROUPING OF PARTS TREATED AS AN ENTITY IN TASK PLANNING

Figure 3 - World knowledge base structure and term definitions

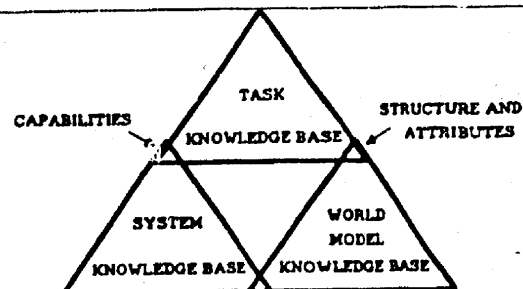


Figure 4 - Knowledge base conceptualization

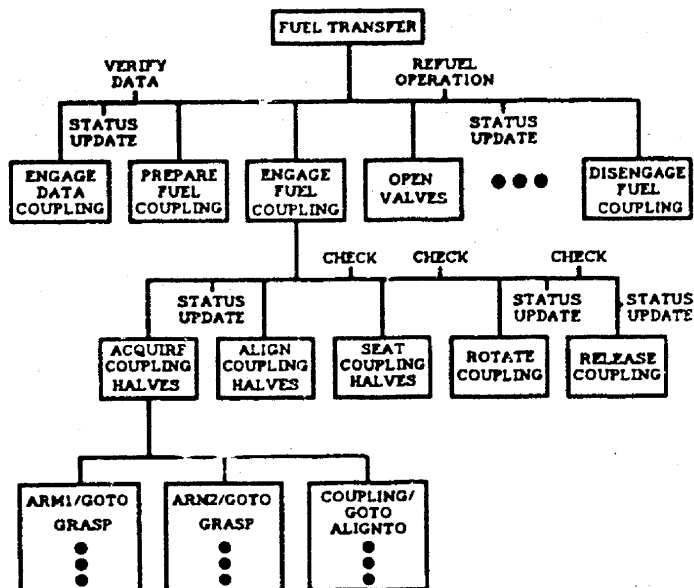


Figure 5 - Task decomposition display

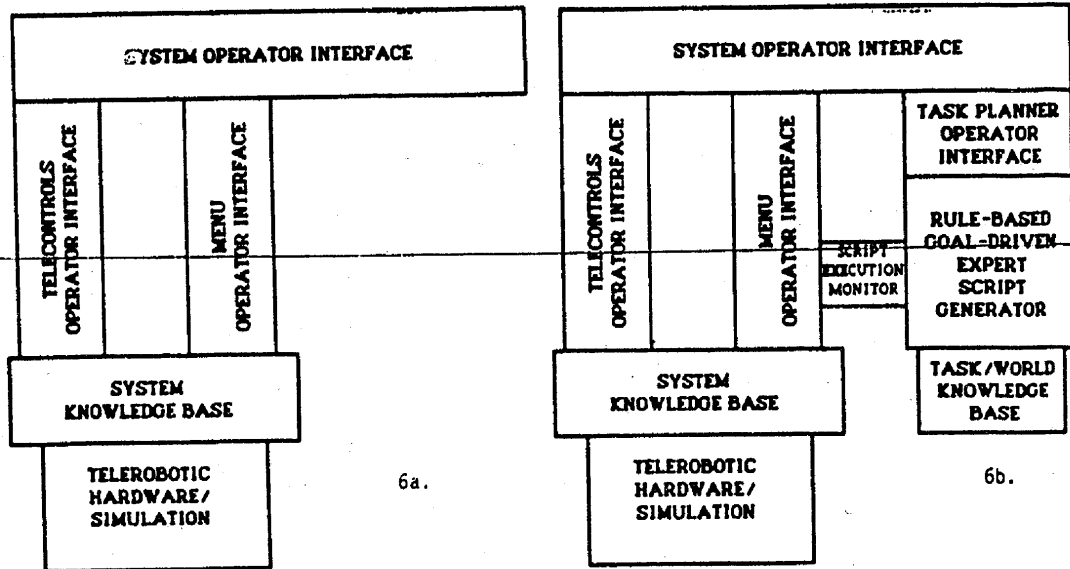


Figure 6 - Rule-based planner structure